An Agent Driven Human-centric Interface for Autonomous Mobile Robots

Donald Sofge, Dennis Perzanowski, Magdalena Bugajska,
William Adams, Alan Schultz
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, DC 20375

ABSTRACT

One of the challenges in implementing a dynamically autonomous mobile robot is achieving a truly human-centric multimodal interface so that human operators can interact with the robot as naturally as they would with another human. Multiple artificial intelligence techniques may be integrated in a distributed computing system through use of an agent-based architecture. In this effort we utilize an agent-based architecture to achieve a multimodal human-centric interface for controlling a dynamically autonomous mobile robot. Capabilities provided by the architecture include natural language understanding, gesture understanding, localization and mapping for accurate navigation, and use of sensors and maps for spatial reasoning.

Keywords: Human-centric, Multimodal, Dynamic Autonomy, CoABS Grid, Mobile Robots

1. INTRODUCTION

One of the challenges in implementing dynamically autonomous behaviors in mobile robots is achieving a truly human-centric interface so that human operators can interact with the robots as naturally as they would with another human. In this effort we facilitate the natural interaction between humans and robots through use of a multimodal interface [1]. We define "human-centric" as focusing on the needs and natural modes of interaction of the human rather than the robot. A key feature of this interface is the use of multiple overlapping modes of communication between the operator and the robot. These overlapping (and sometimes redundant) modes of communication provide the operator with a natural interface to the system, allowing the operator to choose the mode of communication most comfortable to him/her given the current task, situation and environmental conditions. Redundancy in communications is also beneficial in case of subsystem failures, and may also serve to help resolve ambiguity, such as when a command is communicated in more than one way to the robot (e.g. verbally and through a gesture).

Agents provide a natural and flexible means for integrating multiple interface modules together as in our multimodal interface, especially on a distributed computing platform such as an autonomous mobile robot which employs several networked computers and numerous sensors.

The Defense Advanced Research Projects Agency (DARPA) Control of Agent Based Systems (CoABS) Grid architecture provides a flexible and scalable infrastructure for implementing agent-based services in a distributed computing environment. In this effort we utilize agents running on the CoABS Grid in order to demonstrate the effective use of the agent- and grid-based computing on autonomous mobile robots. The agents provide some of the functionality of the interface.

2. MULTIMODAL INTERFACE

Figure 1 shows our human-centric multimodal interface for autonomous mobile robots.

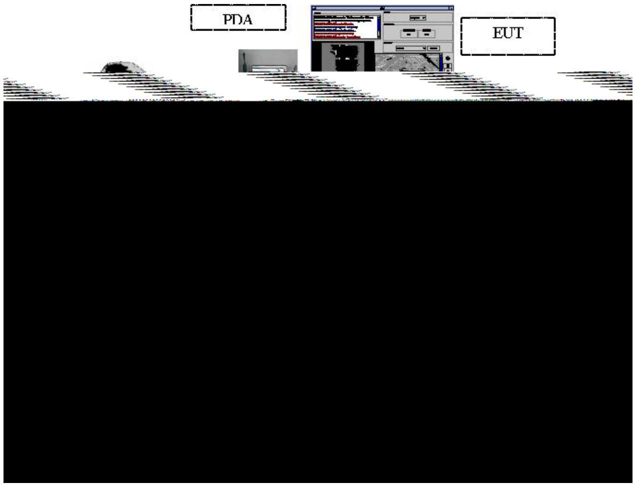

Figure 1 – Human-centric Multimodal Interface

Using this interface, commands may be communicated to the robot in a variety of human-centric ways including verbally, through touch (on a PDA, touch tablet or keyboard), and through gesturing with hands and arms. The robot passes a variety of information back to the operator such as sensor readings, video, navigation maps built using its array of on-board sensors [2], and the status of commands sent to it.

Dynamic Autonomy

Dynamic autonomy allows the robot to dynamically adjust its behaviors depending upon and appropriate to the task(s) at hand [3]. The human operator is able to interact with the robot in a human-centric manner by providing verbal commands and gestures to the robot to perform tasks requiring varying levels of human interaction. Some circumstances may require very fine-grained level operator control, while others may require less precision. Dynamic autonomy used in mobile robots provides a more flexible and operator-friendly interface and makes the robots more versatile.

We support dynamic autonomy in our system through a number of robot behaviors of varying complexity including collision-free navigation, path following, exploration, automatic prioritization of multiple command directives, and feedback from the robot to the operator. Feedback is provided by voice synthesis and through text strings requesting clarification if the robot isn't able to understand the command(s). Operation of the natural language interface with the gesture interpretation process and other command input modes is discussed in greater detail in the section describing the robot's integrated goal-driven architecture.

Understanding Gestures

One of the key modes of interaction with the robot is through the use of gestures. Several types of gestural interfaces have been developed and used in the past. For example, one gestural interface uses stylized gestures of arm and hand configurations ("natural" gestures) [4], while another is limited to the use of gestural strokes on a PDA display ("synthetic" gestures) [5]. In our system we combine both of these approaches, allowing both "natural" and "synthetic" gestures. Our natural gesture interface utilizes a structured-light rangefinder to detect the positions of the hands over several consecutive frames to generate trajectories for the gesture command. The structured-light rangefinder emits a horizontal plane of laser light. A camera mounted on the robot just above the laser is fitted with an optical filter which is tuned to the frequency of the laser. The camera registers the reflection of the laser light off of objects in the room and generates a depth map (XY) based upon location and pixel intensity. The data points for bright pixels (indicating closeness to the robot) are clustered. If a cluster is significantly closer to the robot than background scenery, it is interpreted as being a hand. Hand locations are stored from several consecutive frames, and the positions of the hands are used to generate trajectories for the gesture command. Each trajectory is analyzed to determine if it represents a valid gesture. The command corresponding to the matched gesture is then queued so that the multimodal interface, upon receiving another command, can retrieve the gesture from the gesture queue and combine it with the verbal command in the command interpretation system.

Natural Language Interface

Our natural language interface combines a commercial speech recognition front-end with an in-house developed deep parsing system [3]. ViaVoice is used to translate the speech signal into text, which is then passed to our natural language understanding system, Nautilus, to produce both syntactic and semantic interpretations. The semantic interpretation, interpreted gestures from the vision system, and command inputs from the computer or other interfaces are compared, matched and resolved in the command interpretation system (Figure 1).

Using our multimodal interface the human user can interact with the robot using both natural language and gestures. The semantic interpretation is linked, where necessary, to gesture information via the Gesture Interpreter, Goal Tracker/Spatial Relations component, and Appropriateness/Need Filter, and an appropriate robot action or response results.

For example, the human user can ask the robot "How many objects do you see?" ViaVoice analyzes the speech signal, producing a text string. Nautilus parses the string and produces a representation something like the following, simplified here for expository purposes.

```
(ASKWH
    (MANY N3 (:CLASS OBJECT) PLURAL)
    (PRESENT #:V7791                                   (1)
        (:CLASS P-SEE)
        (:AGENT (PRON N1 (:CLASS SYSTEM) YOU))
        (:THEME N3)))
```

The parsed text string is mapped into a kind of semantic representation, shown here, in which the various verbs or predicates of an utterance (e.g.   ) are mapped into corresponding semantic classes (   ) that have particular argument structures (,   ); for example "you" is the agent of the   class of verbs in this domain and "objects" is the theme of this verbal class, represented as "N3"—a kind of co-indexed trace element in the theme slot of the predicate, since this element is fronted in English wh-questions. If the spoken utterance requires a gesture for disambiguation, as in for example the sentence "Look over there," the gesture components obtain and send the appropriate gesture to the Goal Tracker/Spatial Relations component which combines linguistic and gesture information.

Spatial Reasoning

Spatial reasoning is an important element of a human-centric interface because humans often think in terms of relative spatial positions, and use such relational linguistic terminology naturally in communicating with one another. Our spatial reasoning component builds upon an existing framework of natural language understanding with semantic interpretation [6], and utilizes on-board sensors for detecting objects and map-building through use of evidence grids.

Understanding spatial linguistic terms allows for more efficient and natural control of a dynamically autonomous mobile robot. For example, we may want to give the robot a command such as "Go down the road 100 feet, turn right behind the building and proceed ahead 20 feet. Then go into active surveillance mode." Or, in an office setting, "Go between the table and the chair, through the doorway, and down the hall to the left 50 feet." Spatial reasoning increases the dynamic autonomy of the system by giving the operator a less restrictive vernacular for commanding the robot.

The spatial reasoning component of the multimodal interface allows the robot to provide feedback to the human operator using natural spatial terminology. The human is able to query the robot about the relative spatial positions of objects in the environment, and the robot is able to respond using spatial terms. This is demonstrated in the following dialogue.

Human:	"Tell me what you see."
Robot:	"I see 3 objects."
Human:	"Where are they located?"
Robot:	"Object A is 5 feet in front of me. Object B is 10 feet in front of me and to my right. Object C is 20 feet to my left."

This natural spatial language is used to disambiguate spatial references by both humans and robots [6]. It provides a common interpretation for location expressions, such as "left" and "right", as well as other relative directions. For example, if the human commands the robot, "Turn left," the robot must understand whose left is being referred to, the human's or the robot's. Use of spatial language between humans and robots is currently under investigation by our group at NRL through human-factors experiments [7] where novice users provide instructions to the robot for performing various tasks where spatial referencing is required. This work will result in development of a common language for spatial referencing geared to the needs and expectations of untrained and non-expert operators. This common spatial language will be incorporated into the multimodal interface.

In this effort we focused on the use of agent technology (as implemented under the CoABS Grid) with a highly distributed AI system operating in real-time. A robot control interface was implemented as a CoABS Grid-based agent so that a client located anywhere on the grid could access and control the robot. It was designed such that information (including video) would flow in real-time from the robot through the to the , and commands would flow from the through the to the robot. We also demonstrated that the robot (or robot operator through the robot) could easily access other grid-based services, including services which gather and utilize information from the .

3. MOBILE ROBOT INTEGRATED GOAL-DRIVEN ARCHITECTURE

Figure 2 shows our mobile robot integrated goal-driven architecture. This architecture is organized around providing integration and arbitration for goals presented though various interface modules. Outputs for speech recognition, natural language understanding, gesture interpretation, and other interface modules are cached; command prioritization and resolution are then performed.

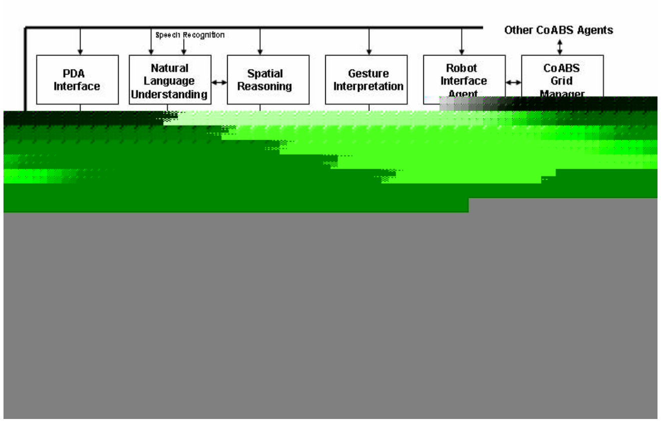

Figure 2 – Mobile Robot Integrated Goal-Driven Architecture

Once goals are interpreted and resolved, they are passed to the routines, where they are integrated with low-level behaviors such as obstacle avoidance, exploration and path planning using the Vector Field Histogram (VFH) method [8]. The architecture maintains both short-term and long-term maps (not shown in Figure 2), which are also important for several of the other processes such as , , and .

The provides a portal for integrating additional capabilities within the architecture. The coordinates all activities over the by allowing agents to register and advertise their services, request services from other agents, and transfer information over the grid to fulfill requests.

The interface between the and the module is implemented as an agent, the . The system includes two CoABS Grid-based services, a which implements a local or remote screen-driven interface to the robot, and a which provides a connection between the robot and the in order to access real-time weather service information.

The enhances the scalability of the architecture by providing a means to add new capabilities easily either directly through the use of other CoABS Grid agents, or through software agents and services accessible via the .

Use of a Central, Unifying Representation

Achieving a robust yet scalable architecture for autonomous mobile robots requires the use of a common representation for integrating motion planning and navigation. The unifying representation used in this work is the evidence grid [9].

Evidence grids provide a probabilistic representation of Cartesian space by dividing the space of the robot into a grid of cells. A real-valued number in the range (-1, 1) is used to express the probability of an individual cell being occupied. The number (1) indicates that a cell is occupied, while the number (-1) indicates that the cell is unoccupied. Under our implementation the evidence grid is populated based upon the returns from the robot’s sensors, including sonar sensors and a planar structured-light sensor. Whereas the sonar sensors are better at providing evidence that an area is empty, the structured light sensor is better at providing evidence that an area is occupied due to its planar 2D nature. Observations are accumulated and the evidence grid is updated using a Bayesian update rule.

As mentioned previously, information is maintained in short-term and long-terms maps. The short-term map shows what the robot immediately senses in its spatial environment, whereas the long-term map is built up over time. The short-term map is used to update the long-term map. Our use of the evidence grid as a common representation is described in detail in [2].

In the next section we briefly describe the CoABS project and the CoABS Grid architecture, including the need for the Grid by the U.S. military and the capabilities of the Grid for designing and implementing agent-based systems. Section 6 details our use of the grid architecture in designing and using the two interface agents mentioned previously in a real-time autonomous mobile robot environment.

4. CONTROL OF AGENT-BASED SYSTEMS (COABS)

The CoABS Grid emerged from the DARPA Information Processing Technology Office (IPTO) program under which this effort was funded. The purpose of CoABS was to foster development and encourage the use of agent-based systems to improve military command, control, communications and intelligence gathering (C^3I). The primary emphasis in CoABS has been the development of a prototype middleware, the CoABS Grid, for coordinating and managing large numbers of cooperating agents over a heterogeneous, loosely coupled network. The CoABS Grid integrates heterogeneous agents, object-based applications, and legacy applications into a common framework whereby agents can register their services dynamically, advertise their capabilities, search for needed services or capabilities, and transmit and receive messages between agents.

The grid is important to the military for potential use with mobile autonomous robotic systems because it provides a logical, highly structured backbone for coordinating these assets and performing battlefield C^3I. This work is one of the first to successfully demonstrate a real-time grid-enabled dynamically autonomous mobile robot which can be supervised from a grid-enabled client located anywhere on the grid, and feedback information from the robot (including video) to the remote operator. It can also be controlled by a local operator via a PDA, voice commands or gestures.

In this effort we utilize two grid-based agents, a for providing a multi-modal interface to the robot, and a for accessing real-time weather information from the web. This is illustrated in Figure 3. Access to each agent module is accomplished through use of a module. The can be dynamically configured to control a variety of different robots by selecting them by name. However, the user can only interact with the single currently selected robot. We are working on scaling up the system to direct multiple robots simultaneously (at a team level) from a single interface.

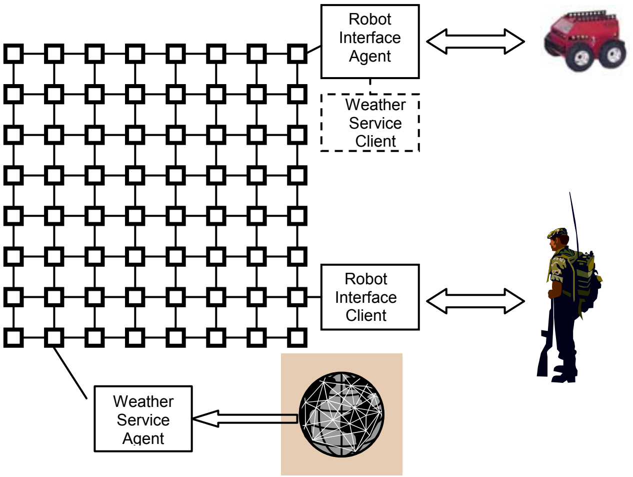

Figure 3 – CoABS Grid Architecture for Dynamically Autonomous Mobile Robots

5. COABS AGENTS FOR AN AUTONOMOUS MOBILE ROBOT

Services over the grid are initiated as follows. First, a is registered with the Grid Manager and advertises its services for providing an interface to the robots. An operator located somewhere on the grid may then start a which identifies and begins a dialogue with the , which then creates a local GUI for the operator to interact with the robots. Once the interface is available, the operator will begin receiving information from the robot including video feed, status information, etc. The operator, through the client, may also begin issuing commands for the robot to perform certain actions. The is shown in Figure 4.

A is registered and advertises its services for providing real-time weather forecasts for cities distributed around the world. The weather agent retrieves and parses html from web pages provided by the National Weather Service. The starts a so that the operator can retrieve weather forecasts through the robot interface. We currently use speech recognition and natural language parsing modules to let an operator make a verbal request for weather information to the robot. The robot “understands” that request and submits the weather request through the to the . The , which has a web connection, retrieves the appropriate web page and parses it to extract the desired weather report. The weather report is sent through the to the (where it is displayed on the GUI of the client computer) and is also sent to the robot, where it is read back to the operator using the on-board voice synthesizer.

The GUI (Figure 4) is used to control the robot as well as present the current status of the robot and other information useful to the operator, such as weather reports. The message window in the upper left corner (label 1) displays the weather reports and status information on tasks assigned to the robot, such as when each task has been completed. The messages are color-coded according to the priority level of the message.

The robot selector window to the right of the message window (label 2) shows the robot currently being controlled. The operator can switch between robots using the pull-down menu. The window below the robot selector window (label 3) shows meteorological data collected by the robot, currently wind speed and direction.

The weather request window (label 4) allows the operator to select a city via a pull-down menu and then by clicking <Submit> to send a request to the
for a weather report. As discussed previously the operator may also get the weather report by a making verbal request to the robot. Once retrieved the weather report is displayed in the message window and it is also spoken by the robot.

Figure 4 – Robot Interface GUI Client

The map window (label 5) is used to show the robot’s current map of its environment. Maps may be built up through exploration of the environment, stored, and later reloaded. The map window also serves as a command input mechanism for the operator to direct the robot to a specific location on the map, or to trace a path on the map for the robot to follow. The operator may use a mouse (or touch screen on the PDA) to indicate the goal point or path on the map. A trace of the recent path of the robot can be overlaid onto the map to provide additional information.

The large window to the right of the map (label 6) shows an aerial (e.g. satellite) image of an area, and is used in conjunction with the robot's on-board GPS navigation system for outdoor navigation. The operator can pan and zoom the overhead image, and as with the map window the operator can direct the robot to a specific location by clicking on the map or have it follow a path by tracing it on the map. As in the map window, a trace of the recent path of the robot can be overlaid onto the image map.

In the bottom left corner of the GUI (label 7) the video window shows the current video feed from the robot. This provides the operator with a remote view of what the robot is "seeing", with a video frame refresh rate sufficient for teleoperation of the robot using the joystick interface, accessed by clicking on the <Joystick> button (label 8).

6. CONCLUSIONS

The agent-based architecture provides a natural and highly scalable approach to implementing a human-centric multimodal interface for dynamically autonomous mobile robots. In this effort we demonstrate the use of agent technology to expand the capabilities of our multimodal interface. The demonstrates that a grid-enabled robot can utilize an external agent which accesses the web in order to retrieve potentially important information. Additional agents could be implemented which take advantage of the grid architecture, both enhancing the capabilities of the robot directly, and providing access to the operator in the field for services through the robot (such as the weather).

We are currently expanding the capabilities of our robots in a variety of ways including additional spatial reasoning to allow the robots to make decisions with regard to objects in their immediate environment, adding a cognitive architecture to allow the robots to reason in a more human-like manner, and enhancing the multimodal interface to allow a single operator to control multiple robots simultaneously (robot teams).

7. ACKNOWLEDGEMENTS

Support for this effort was provided by the Defense Advanced Research Projects Agency IPTO CoABS Program under MIPR 02-M733-00.

8. REFERENCES

[1] Perzanowski, D., Adams, W., Schultz, A., and Marsh, E. (2000). "Towards Seamless Integration in a Multimodal Interface", **Proceedings 2000 Workshop Interactive Robotics and Entertainment**, pages 3-9, Menlo Park, CA.

[2] Schultz, A., Adams, W., and Yamauchi, B. (1999). "Integrating Exploration, Localization, Navigation and Planning Through a Common Representation", **Autonomous Robots**, 6(3), Kluwer.

[3] Perzanowski, D., Schultz, A., Adams, W., and Marsh, E. (1999). "Goal Tracking in a Natural Language Interface: Towards Achieving Adjustable Autonomy", In **Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation**, Monterey, CA.

[4] Kortenkamp, D., Huber, E. and Bonasso, P. (1996). "Recognizing and Interpreting Gestures on a Mobile Robot", In **Proceedings of *AAAI***, 1996.

[5] Fong, T. W., Conti, F., Grange, S. and Baur, C. (2000), "Novel Interfaces for Remote Driving: Gesture, haptic, and PDA", **SPIE 4195-33, SPIE Telemanipulator and Telepresence Technologies VII**, Boston, MA.

[6] Skubic, M., Perzanowski, D., Schultz, A., and Adams, W. (2002). "Using Spatial Language in a Human-Robot Dialog", In **Proceedings 2002 IEEE Conference on Robotics and Automation**, IEEE.

[7] Perzanowski, D., Brock, D., Blisard, S., Adams, W., Bugajska, M., Schultz, A., Trafton, G., Skubic, M. (2003), "Finding the FOO: A Pilot Study for a Multimodal Interface", In **Proceedings of the IEEE Systems, Man, and Cybernetics Conference**, Washington, DC.

[8] Borenstein, J. and Koren, Y. (1991). "The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots", **IEEE Transactions on Robots and Automation,** IEEE: New York.

[9] Moravec, H. and Elfes, A. (1985). "High Resolution Maps from Wide Angle Sonar", In **Proceedings of the IEEE International Conference on Robotics and Automation**.